Practical 7: Write functions to find FIRST and FOLLOW of all the variables.

Aim: Write a suitable data structure to store a Context Free Grammar. Prerequisite is to eliminate left recursion from the grammar before storing. Write functions to find FIRST and FOLLOW of all the variables. [May use unformatted file / array to store the result].

Algorithm:

First ()-

If x is a terminal, then FIRST(x) = { 'x' }

If x-> Є, is a production rule, then add Є to FIRST(x).

If X->Y1 Y2 Y3….Yn is a production,

FIRST(X) = FIRST(Y1)

If FIRST(Y1) contains Є then FIRST(X) = { FIRST(Y1) – Є } U { FIRST(Y2) }

If FIRST (Yi) contains Є for all i = 1 to n, then add Є to FIRST(X).

Follow ()-

FOLLOW(S) = { $ } // where S is the starting Non-Terminal

If A -> pBq is a production, where p, B and q are any grammar symbols,

then everything in FIRST(q) except Є is in FOLLOW(B).

If A->pB is a production, then everything in FOLLOW(A) is in FOLLOW(B). If A->pBq is a production and FIRST(q) contains Є,

then FOLLOW(B) contains { FIRST(q) – Є } U FOLLOW(A)

Program: